

CLAIMS

Now, therefore, the following is claimed:

- 1 1. A code verification system, comprising:
2 memory for storing a compiled program; and
3 a code verifier configured to analyze instructions of said program and to
4 generate a plurality of type signatures based on said instructions, each of said type
5 signatures indicating each input type constraint and each output type description for a
6 respective one of said instructions, wherein said code verifier is configured to detect a
7 type error by analyzing said type signatures.
- 1 2. The system of claim 1, wherein said code verifier is configured to
2 compose one of said type signatures with another of said type signatures to form a
3 single composed signature, said code verifier further configured to make a
4 determination as to whether an input type constraint of said one type signature is
5 acceptable to an output type description of said other type signature, said code verifier
6 further configured to detect said type error based on said determination.
- 1 3. The system of claim 1, wherein said code verifier is further configured
2 to analyze said program and to group instructions of said program into a plurality of
3 code blocks, wherein said code verifier is configured to translate said code blocks into
4 type signature blocks, each of said type signature blocks having one or more type
5 signatures, said code verifier further configured to compose the type signatures of each
6 of said type signature blocks into a single respective composed type signature.

1 4. The system of claim 3, wherein said code verifier, in composing one of
2 said type signature blocks, is configured to compose a first type signature with a
3 second type signature to form a single composed signature, said first and second type
4 signatures included within said one type signature block, said code verifier further
5 configured to make a determination as to whether an input type constraint of said first
6 type signature is acceptable to an output type description of said second type
7 signature, said code verifier further configured to detect said type error based on said
8 determination.

1 5. A code verification system, comprising:
2 memory for storing a compiled program; and
3 a code verifier configured to analyze a code block of said program and to
4 translate instructions within said code block into a plurality of type signatures, said
5 code verifier further configured to compose said type signatures into a single
6 composed type signature and to detect a type error by analyzing said type signatures.

1 6. The system of claim 5, wherein one of said type signatures includes a
2 type description indicative of a type of an input consumed by one of said instructions
3 when said one instruction is executed, wherein another of said type signatures includes
4 a type description indicative of a type of an output produced by another of said
5 instruction when said other instruction is executed, and wherein said code verifier is
6 further configured to detect said type error by comparing said type descriptions.

1 7. The system of claim 5, wherein said code verifier is further configured
2 to analyze said program and to group instructions of said program into a plurality of
3 code blocks, wherein said code verifier is configured to translate said code blocks into
4 type signature blocks, each of said type signature blocks having one or more type
5 signatures, said code verifier further configured to compose the type signatures of each
6 of said type signature blocks into a single respective composed type signature.

1 8. A code verification method, comprising the steps of:
2 storing a compiled program;
3 generating a plurality of type signatures based on instructions within said
4 program, each of said type signatures indicating each input type constraint and each
5 output type description for a respective one of said instructions;
6 analyzing said type signatures; and
7 detecting a type error based on said analyzing step.

1 9. The method of claim 8, further comprising the steps of:
2 composing one of said type signatures with another of said type signatures
3 thereby forming a single composed signature; and
4 determining whether an input type constraint of said one type signature is
5 acceptable to an output type description of said other type signature,
6 wherein said detecting step is further based on said determining step.

1 10. The method of claim 8, further comprising the steps of:
2 grouping instructions of said program into a plurality of code blocks;
3 translating said code blocks into type signature blocks, each of said type
4 signature blocks having one or more type signatures; and
5 composing the type signatures of each said type signature blocks into a single
6 respective composed type signature.

1 11. A code verification method, comprising the steps of:
2 storing a compiled program, said compiled program having at least one code
3 block that includes a plurality of instructions;
4 translating said instructions into a plurality of type signatures;
5 composing said type signatures into a single composed type signature;
6 analyzing said type signatures; and
7 detecting a type error based on said analyzing step.

1 12. The method of claim 11, wherein one of said type signatures includes a
2 type description indicative of a type of an input consumed by one of said instructions
3 when said one instruction is executed, wherein another of said type signatures includes
4 a type description indicative of a type of an output produced by another of said
5 instructions when said other instruction is executed, and wherein said analyzing step
6 includes the step of comparing said type descriptions.

- 1 13. The method of claim 11, further comprising the steps of:
- 2 grouping instructions of said program into a plurality of code blocks;
- 3 translating said code blocks into type signature blocks, each of said type
- 4 signature blocks having one or more type signatures; and
- 5 composing the type signatures of each of said signature blocks into a single
- 6 respective composed type signature.